УДК 004.9

С.М. ЛИСЕНКО, А.Ф. КРИЩУК
Хмельницький національний університет
Ю.П. ДЗЮБАК
Нововолинський електромеханічний коледж

# ДОСЛІДЖЕННЯ ПЕРЕВАГ ЗАСТОСУВАННЯ LATEX ПРИ ОФОРМЛЕННІ НАУКОВИХ ПРАЦЬ

*В роботі досліджено актуальність застосування мови розмітки даних та пакетів макросів TeX для високоякісного оформлення документів. В роботі також подано основні прийоми створення наукового TeX документу. В статті представлені якісні переваги використання системи LATEX як засобу створення математичних і технічних текстів для публікації в наукових виданнях.*

*In the article the urgency of the using of a typesetting system and TeX macro package for high quality documentation is researched. Also in the paper a scientific techniques of the TeX document creating is presented. The article presents advantages of LATEX system as a means of creating mathematical and technical texts for publication in scientific journals.*

Ключові слова: TEX, LATEX, Springer, IEEE, IEEE data Xplore, TEX-верстка, стилі в TEX.

## Вступ

Складність процесу підготовки матеріалів до видання, яка пов’язана з некомпетентністю і неуважністю користувачів при ознайомленні з матеріалами вимог створює проблему і незручності як для редакторів, так і для замовників та користувачів. Використання текстового процесора Latex дозволяє позбутись більшості даних проблем і прискорити процес виходу друкованих видань.

Латекс являє собою набір програм і є продовженням оригінальної програми TEX, написаної Дональдом Кнутом [1].

У більшості текстових процесорів операції введення тексту, форматування рядків, абзаців і сторінок, виведення редагованого тексту на екран, а згодом на принтер, об'єднані в одному додатку. Програма верстки TEX відповідає тільки за форматування рядків, абзаців і сторінок. Таким чином, набір документу, використовуючи TEX, здійснюється введенням тексту документа і необхідних команд форматування в текстовому редакторі (наприклад, блокнот в Windows), а потім його компіляцією. Після цього документ можна переглянути за допомогою попереднього перегляду або друку з використанням драйвера принтера.

TEX також є мовою програмування, що дозволяє написання коду для додаткових функцій, що забезпечує зручне використання даного продукту для написання наукових робіт різного спрямування.

Сьогодні найбільш відомі видавництва приймають матеріали лише у форматі Latex, зокрема міжнародна видавнича компанія, що спеціалізується на видавництві академічних журналів та книг з природничо-науковим спрямуванням Springer [2].

## Структура LATEX документу

Лістинг LATEX документу з підтримкою української мови:

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\begin{document}
\begin{theses}{ukrainian}
Це мій \emph{перший} документ в \LaTeX.
\end{theses}
\end{document}
```

Для роботи з даним документом необхідно зберегти файл на жорсткому диску з іменем `file.tex`, або з іншим іменем, яке дозволене операційною системою. Подальші дії користувача залежать від типу операційної системи [3-4].

## GNU / Linux

У командному рядку необхідно ввести:

```
latex myfile
```

На екрані видно кількість рядків прокручування тексту. Для перегляду результатів у вікні терміналу необхідно ввести наступну стрічку:

```
xdvi myfile
```

З'явиться вікно, яке показує результат зображений на рисунку 1.

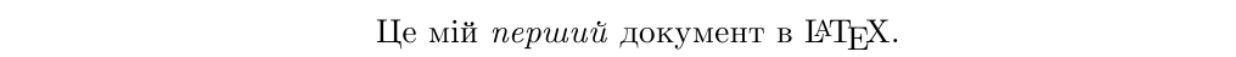


**Рис.1. Результат виводу стрічки тексту із стилізованим позначенням**

Перший рядок `\documentclass{article}` повідомляє LATEX, що результатом буде стаття. Для написання книг, або інших підготовлених матеріалів, параметр `article` має бути змінено на відповідний тип. Весь текст документу, який буде набрано, повинен бути розміщеним між командами `\begin{document}` і `\end{document}`.

З 1998 року діє офіційний стандарт з підтримки кирилиці в TeX/LaTeX (включно з українською мовою). Він включає підтримку кирилиці в LaTeX 2ε (починаючи з версії від 31 грудня 1998 р.), підтримку української мови в `babel`, кириличні LH шрифти і українські переноси. Команди `\begin{theses}{ukrainian}` і `\end{theses}` забезпечують можливість введення українських літер за наявності відповідного стилю в поточному каталозі документу.

З рисунку 1 видно, що перші два слова виводяться як типізовані. У позиціях тексту, де `\emph{перший}` є командою LATEX, виконується верстка тексту, а у фігурних дужках в курсив («`italic`»). Команда `\LaTeX` виводить у вихідний файл стилізоване позначення LATEX [2,5-8].

Основне призначення TEX - це вірно відформатований технічний документ, який може містити багато математичних описів. Виправляти складні математичні формули, використовуючи текстовий процесор LaTeX, маючи початковий лістинг, досить просто.

**Верстка документу**

Для створення латекс-документу текст друкується разом з деякими командами LATEX. Слова мають бути розділені пробілами (не важливо скільки).

Для створення нового абзацу потрібно залишити порожній рядок.

Хоча довжина рядка тексту, що вводиться, буває різною, у вихідному файлі вона є однаковою. Зміна відстані між словами здійснюється системою TEX.

Можлива ситуація, коли необхідний додатковий відступ після закінчення речення, наприклад:

```
В кінці абзацу задається порожній рядок на вході. Тобто для початку нового абзацу, потрібно залишити порожній рядок.
```

Лістингом даного тексту буде:

```
В кінці абзацу задається порожній рядок на вході\@. Тобто для початку нового абзацу, потрібно залишити порожній рядок.
```

Команди `\@` використовуються перед крапкою для створення додаткового відступу після закінчення речення.

Приклад випадку, коли речення після знаку «.» (крапка) продовжується з малої літери:

```
Робота була виконана д.т.н., проф. Петренком П.П.
```

Лістингом даного тексту буде:

```
Робота була виконана д.\ т.\ н.\ , проф.\ Петренком П.П.
```

В даному випадку використовується команда \ (зворотній «слеш» і пробіл). Дана команда також використовується для позначення пробілу у вихідному коді.

Можливі ситуації, коли команда \ є корисною для вставки спеціальних команд і спеціальних символів. Наприклад:

```
Я думаю \LaTeX  це цікаво.
```

Результат зображено на рисунку 2.

**Рис. 2. Результат виконання лістингу без використання команди «\»**

Як видно з рисунку 2, TEX усуває всі пробіли після команд. Для отримання потрібної послідовності у вихідному файлі необхідно змінити вхідну послідовність на наступну:

```
Я думаю \LaTeX\ це цікаво.
```

Використання лапок у лістингу LaTeX відрізняється від загально відомого.

Результат виконання лістингу TEX:

```
Зверніть увагу на праві і ліві лапки ‘одинарні лапки’ і “подвійні лапки”.
```

Це описується наступним лістингом:

```
Зверніть увагу на праві і ліві лапки ‘одинарні лапки’ і ‘подвійні лапки’’.
```

Сучасні клавіатури містять клавішу для введення символу `, яка інтерпретується як ліва лапка TEX. (У наведеному моделюванні вхідного лістингу, цей символ відображено як ‘). Крім того, клавішу ‘інтерпретує ліва лапка в TEX. Подвійні лапки компілюються шляхом введення одинарних лапок двічі. ‘ Також ‘ подвійні лапки ‘‘ можуть бути використані для закриття подвійних лапок в TEX [3, 9-13].

Якщо на клавіатурі немає клавіші лівих лапок, можна використовувати команду \lq для вставки їх у текст. Відповідна команда \rq встановлює праві лапки. Таким чином, наведений вище текст може бути виконано наступним чином:

```
Зверніть увагу на праві і ліві лапки \lq одинарні лапки \rq\ і \lq\lq
```

`подвійні лапки \rq\rq.`

Нові версії компіляторів допускають використання відкриваючих і закриваючих подвійних лапок у лістингу для відображення їх у вихідному тексті.

Символи тире у тексті використовуються для різних цілей. У TEX тире мають різну довжину. Таким чином, короткі тире використовуються для дефісів, більші використовуються для позначення діапазонів номерів і ще більші тире використовуються для зауважень та пояснень. Приклад використання тире в TEX, вихідний текст:

```
Троянські програми розглядаються на сторінках 110-150 том 5—захист комп'ютерних мереж.
```

Лістинг TEX:

```
Троянські програми розглядаються на сторінках 221--225 том 5--- захист комп'ютерних мереж.
```

Один символ тире у лістингу - компілює перенос слова у вихідному файлі, два тире -- компілює більше тире (–) і три тире --- компілює найдовше довге тире (-) на виході.

При наборі іноземних слів, можливе використання позначень над літерами різних типів. На рисунку 3 приведені позначки доступні в LATEX.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ò | \‘o | ó | \’o | ô | \^o | õ | \~o |
| ō | \=o | ȯ | \.o | ö | \"o | ç | \c c |
| ŏ | \u o | ǒ | \v o | ő | \H o | ọ | \d o |
| o̲ | \b o | o͡o | \t oo | | | | |

**Рис. 3. Спеціальні позначення в LATEX для роботи з іноземними словами**

Букви `i` та `j` можуть потребувати спеціального відображення і не повинні мати свої звичайні точки. Команди `\i` і `\j` відображають менші точки, ніж стандартні [14-15]. Таким чином, щоб отримати результати, що вказані на рисунку 4, необхідно написати:

```
\'{E}l est\'{a} aqu\'{\i}
```

Él está aquí

**Рис. 4. Результат виконання коду**

Деякі символи, які характерні для англійської мови, також доступні в LATEX, як показано на рисунку 5.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| œ | \oe | Œ | \OE | æ | \ae | Æ | \AE |
| | \aa | | \AA | | | | |
| ø | \o | Ø | \O | ł | \l | Ł | \L |
| ß | \ss | | | | | | |
| ¡ | !‘ | ¿ | ?‘ | | | | |

**Рис. 5. Спеціальні символи в LATEX**

Значення `\LaTeX` виводить стилізований у емблему LATEX у вихідний файл, а символ \ виводить пробіл. TEX використовує символ \ для спеціальних цілей, щоб вказати програмі, що далі слідує не набраний текст, а команда для виконання. Тому, якщо необхідно отримати \ у вихідному файлі, потрібно ввести команду `\textbackslash` , результатом виконання якої і буде \ .

Таким чином, \ є символом, який має особливе значення для TEX, і він не може бути отриманий шляхом прямого введення [1, 4,17]. Також можлива наступна ситуація:

```
Я щось дізнався про % \ LaTeX.
```

Результатом буде наступний текст:

```
Я щось дізнався про
```

TEX використовує в символ % відсотків як символ коментаря. TEX розглядає наступний текст як "коментар", а не як набраний текст. Це корисно для TEX-програміста, щоб пояснити особливості спеціалізованих частин коду.

Для виводу знаку відсотка в лістингу потрібно ввести команду \% .

```
Я щось дізнався про \% \LaTeX.
```

Повний перелік зарезервованих символів системи TEX наступний:

```
~ # $ % ^ & _ \ { }
```

Для відображення цих символів у вихідному документі необхідно ввести комбінації, які наведені в таблиці 1.

Таблиця 1.

**Результати введення команд у TEX документ**

| **Вихідне значення** | **Вхідне значення** | **Вихідне значення** | **Вхідне значення** |
|---|---|---|---|
| ~ | \textasciitilde | \ | \textbackslash |
| & | \& | % | \% |
| # | \# | { | \{ |
| _ | _ | } | \} |
| $ | \$ | ^ | \textasciicircum |

Згідно таблиці 1 більшість символів відображаються з додаванням символу \. Для символів ~ \ та ^ використовуються спец команди [18-20]. Два символи \\ створюють обрив рядка, наприклад:

```
Це перша лінія.\\ Це друга лінія
```

Результат зображений на рисунку 5.

Це перша лінія.
Це друга лінія

**Рис. 5. Результат виконання лістингу**

Також можна змінювати аргумент команди \\ для збільшення відстані між рядками. Наприклад:

```
Це перша лінія.\\[15] Це друга лінія
```

Результат зображений на рисунку 6.

Це перша лінія.

Це друга лінія

**Рис. 6. Результат виконання коду**

На рисунку 6 показано наявність додаткових 15 точок між рядками (1 точка має розмір 1/72 дюйма).

З наведених вище прикладів видно, що TEX вирівнює текст по-своєму, незалежно від способу форматування тексту в початковому файлі. Якщо необхідно створити форматований лист, як зображено на рисунку 7, то необхідно ввести наступний лістинг:

```
\begin{flushright}
Директору ТзОВ ``Програмування''\\
Петренку П.П.\\
Провідного інженера \\
Степаненка С.С.\\[.75cm]
\end{flushright}
\begin{center}
Заява\\[.75cm]
\end{center}
\noindent Прошу Вас надати мені відпустку з 1.09.2012 р. на 30 календарних днів.\\[.75cm]
15.08.2012 р. \qquad С. С. Степаненко
```

Директору ТзОВ “Програмування”
Петренку П.П.
Провідного інженера
Степаненка С.С.

Заява

Прошу Вас надати мені відпустку з 1.09.2012 р. на 30 календарних днів.

15.08.2012 р. С. С. Степаненко

**Рис. 7. Приклад форматованого тексту**

Використана команда `\qquad` визначає пробіл подвійного розміру (2 точки).

Конструкція `\begin{center} ... \end{center}` визначає розміщення тексту точно в центрі сторінки.

Конструкція `\begin{flushright} ... \end{flushright}` визначає розміщення тексту з правого краю сторінки.

Конструкція `\begin{flushleft} ... \end{flushleft}` визначає розміщення тексту з лівого краю сторінки.

Вказані приклади ілюструють LATEX конструкцію, яка називається середовищем. Конструкція має вигляд:

`\begin{`*назва*`} ... \end{`*назва*`}` ,де *назва* є назвою середовища.

В LATEX тип стилю визначається сімейством, серією і формою, які показані в таблиці 2.

Таблиця 2.

**Команди визначення типу стилю в LATEX**

| Тип стилю | Стиль | Команда |
|---|---|---|
| Сімейство | roman<br>sans serif<br>typewriter | \textrm{roman}<br>\textsf{sans serif}<br>\texttt{typewriter} |
| Серія | medium<br>**boldface** | \textmd{medium}<br>\textbf{boldface} |
| Форма | upright<br>*italic*<br>slanted<br>SMALL CAP | \textup{upright}<br>\textit{italic}<br>\textsl{slanted}<br>\textsc{small cap} |

Будь-який тип стилю у вихідному файлі являє собою комбінацію з цих трьох характеристик. За замовчуванням встановлено сімейство `roman`, `medium` серію, та форму `upright` у вихідному LATEX документі. Набір `roman`, `medium`, *`italic`* реалізує команда `\textit`. Набір `roman`, **`boldface`**, `upright` реалізує команда `\textbf`. Є можливість об'єднати ці команди для побудови широкого спектру типів стилів. Наприклад:

```
\textsf{\textbf{sans serif family, boldface series, upright shape}}
\textrm{\textsl{roman family, medium series, slanted shape}}
```

Результат зображений на рисунку 8.

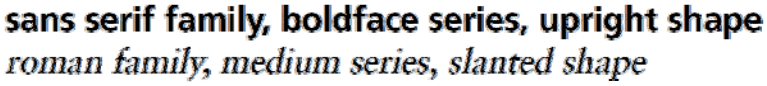


**Рис. 8. Приклад використання стилів**

Деякі з цих типів стилів можуть бути недоступні. У такому випадку LATEX дає попередження під час компіляції і пропонує здійснити заміну іншим доступним типом стилю, який система вважає наближеним до необхідного.

Команда `\emph`. Якщо є звичайний (`upright`) текст, то при використанні команди `\emph` отримуємо курсивний текст. Але якщо поточна форма вказана як `slanted` або *`italic`* , тоді отримуємо текст `upright` форми. Наприклад:

```
\textit{Даний текст надруковано курсивом, але \emph{тепер стандартно} і знову курсив \emph{і стандартно}}
```

Результат:

*Даний текст надруковано курсивом, але* тепер стандартно *і знову курсив* і стандартно

В той же час:

```
\textbf{ Даний текст жирний, але
\emph{ тепер курсивом } і знову жирний
\emph{i курсивом}}
```

Результат:

**Даний текст жирний, але *тепер курсивом* і знову жирний *і курсивом***

Кожна з цих команд зміни типів стилів має альтернативну форму. Наприклад, `\textbf{boldface}` можна також ввести `{\bfseries boldface}`, щоб отримати **boldface**. Команда може виконувати іншу за змістом функцію. Наприклад:

Під **трикутником**, ми розуміємо полігон з трьома сторонами.

Якщо набрати

```
Під \bfseries{трикутником}, ми розуміємо полігон з трьома сторонами.
```

Отриманий результат буде таким:

Під **трикутником, ми розуміємо полігон з трьома сторонами.**

Таким чином, щоб задіяти конкретний фрагмент тексту, декларація та текст повинні бути розміщені в дужках.

| | STYLE | COMMAND | DECLARATION |
|---|---|---|---|
| SHAPE | upright | \textup{upright} | {\upshape upright} |
| | *italic* | \textit{italic} | {\itshape italic} |
| | *slanted* | \textsl{slanted} | {\slshape slanted} |
| | SMALL CAP | \textsc{small cap} | {\scshape small cap} |
| SERIES | medium | \textmd{medium} | {\mdseries medium} |
| | **boldface** | \textbf{boldface} | {\bfseries boldface} |
| FAMILY | roman | \textrm{roman} | {\rmfamily roman} |
| | sans serif | \textsf{sans serif} | {\sffamily sans serif} |
| | typewriter | \texttt{typewriter} | {\ttfamily typewriter} |

**Рис. 9. Команди та декларації для отримання типів стилів**

### Розмір шрифту

Традиційно кегль вимірюється в точках (так званих принтер-точках). За замовчуванням тип TEX має розмір 10 пт. В системі LATEX наявні декларацій для зміни розміру шрифту. Вони приведені в таблиці 3.

Таблиця 3

**Декларації в LATEX для зміни розміру шрифту**

| Результат | Вхідне значення | Результат | Вхідне значення |
|---|---|---|---|
| Розмір | {\tiny розмір} | розмір | {\large розмір } |
| Розмір | {\scriptsize розмір } | розмір | {\Large розмір } |
| Розмір | {\footnotesize розмір } | розмір | {\LARGE розмір } |
| Розмір | {\small розмір } | розмір | {\huge розмір } |
| Розмір | {\normalsize розмір } | розмір | {\Huge розмір } |

Вхідне значення розміру шрифта `\normalsize` відповідає розміру за замовчуванням. Розміри утворюють упорядковану послідовність від найменших `\tiny` до великих `\Huge`.

Допускається об'єднання зміни стилю зі зміною розміру. Наприклад, заява, подана на рисунку 7, може бути відредагована:

```
\begin{flushright}
{\scshape\large Директору ТзОВ ``Програмування''\\
Петренку П.П.\\
Провідного інженера \\
Степаненка С.С.\\[1cm]}
\end{flushright}
\begin{center}
{\bfseries\huge Заява}\\[.75cm]
\end{center}
```

```
\noindent Прошу Вас надати мені відпустку з 1.09.2012 р. на 30 календарних днів.\\[.75cm]
15.08.2012 р. \qquad С.С. Степаненко
```

Скомпільований результат зображений на рисунку 10.

Директору ТзОВ "Програмування"
Петренку П.П.
Провідного інженера
Степаненка С.С.

# Заява

Прошу Вас надати мені відпустку з 1.09.2012 р. на 30 календарних днів.

15.08.2012 р. С.С. Степаненко

**Рис. 10. Приклад використання зміни стилю і розміру шрифтів**

**Оформлення документу**

Виконання роботи у LATEX дає можливість створення глав і розділів у документі. Усі файли LATEX слід розпочинати з вказівки виду документа. Це здійснюється за допомогою команди `\documentclass{...}`. Для коротких статей записується `\documentclass{article}`. Для книги вказується команда `\documentclass{book}`. Також присутні інші класи документів доступні в LATEX, такі як звіт і лист.

LATEX не має класу за замовчуванням . До визначення типу документу можна вказати деякі опції, які модифікують формат за замовчуванням. Синтаксис команди `\documentclass` є наступним:

```
\documentclass [опції] {клас}
```

Опції (варіанти) вказуються в квадратних дужках. Типовим для LATEX-команд є те, що параметри задаються у квадратних дужках, після чого обов'язкові аргументи вносяться у фігурні дужки [21-23].

Текстовий процесор LATEX дає змогу вибрати розмір шрифту для звичайного тексту у всьому документі. Наприклад команди:

```
10pt 11pt 12pt
\documentclass[12pt]{article}
```

встановлюють розмір звичайного тексту в документі із значенням 11 пт. Якщо не вказувати розмір шрифта, то за замовчуванням встановлюється 10пт.

LATEX має власний метод розмітки для створення параграфів. Він також має методи, щоб зробити вертикальні розриви для здійснення виведення різних сторінок. Для застосування обривів компілятору необхідно знати ширину і висоту паперу. Приклад наведено на рисунку 11.

| | | | |
|---|---|---|---|
| letterpaper | 11×8.5 in | a4paper | 20.7×21 in |
| legalpaper | 14×8.5 in | a5paper | 21×14.8 in |
| executivepaper | 10.5×7.25 in | b5paper | 25×17.6 in |

**Рис. 11. Команди встановлення розміру сторінки**

З ростом розмірності вертикалі збільшується висота сторінки. За замовчуванням встановлена розмірність `letterpaper`.

Можливі варіанти для розміщення вмісту кожної сторінки в одну або дві колонки. Це встановлюється наступними опціями:

```
onecolumn   twocolumn
```

За замовчуванням встановлено `onecolumn`.

Існує також можливість вказати, чи буде документ остаточно надрукований тільки на одній стороні паперу чи з обох сторін. Це забезпечується наступними опціями:

```
oneside   twoside
```

У класах «доповідь» і «книга» є положення, щоб вказати різні глави. Глави завжди починаються з нової сторінки. При цьому залишається порожній простір на попередній сторінці, якщо це необхідно. У класі «книга» є додаткові обмеження, які полягають в тому, що глави починаються тільки на непарних сторінках. При цьому, залишається вся порожня сторінка, якщо це буде необхідно. Дані дії контролюються опціями:

```
openany   openright
```

За замовчуванням встановлена опція `openany` для класу `reportclass` і опція `openright` для класу «книга».

В LATEX також існує положення для форматування "заголовків" документа (назва документа, автор(и) та ін.). В класі `article` ця частина документа друкується разом з текстом на першій сторінці, а для доповіді і книги - на окремій сторінці. Ці параметри встановлюються опціями:

`notitlepage titlepage`

За замовчуванням `notitlepage` для статті та `titlepage` для звіту і книги. Як і у випадку інших варіантів, дії за умовчанням можуть бути скасовані явно вказавши в команді опцію `documentclass`.

**Стиль сторінки**

Вказавши загальний тип документа за допомогою команди `\documentclass` з його різними варіантами, можна встановити стиль для окремих сторінок. На мові LATEX кожна сторінка має так звані «голову» і «ногу». Вони містять таку інформацію, як номер поточної сторінки, поточного пункту або розділу. Стиль встановлюється за допомогою команди:

`\pagestyle{...}` , де обов'язковим аргументом може бути один зі стилів, які подано в таблиці 4.

Крім того, можна налаштувати стиль для поточної сторінки за допомогою команди:

`\thispagestyle{`*`style`*`}`

де `style` є назвою одного зі стилів. Наприклад, номер сторінки може бути відсутнім для поточної сторінки за допомогою задання команди `\thispagestyle{empty}`. На наступній сторінці нумерація буде продовжуватись.

У стилі `myheadings`, можна задати текст, який з'явиться у «голові» кожної сторінки. Це робиться однією з команд:

`\markboth{`*`left head`*`{`*`right head`*`}`

`\markright{`*`right head`*`}`

де `left head` – текст, що з'явиться в «голові» лівої сторінки, і `right head` - текст, що з'явиться в «голові» правої сторінки.

Таблиця 4

**Перелік стилів у LATEX**

| Стиль | Опис |
|---|---|
| plain | «Голова» сторінки порожня і «нога» містять лише номер сторінки, в центрі по відношенню до ширини тексту. Це значення за замовчуванням для класу `article`, якщо команда `\pagestyle` не зазначена в преамбулі. |
| empty | «Голова» і «нога» порожні. Зокрема, відсутні номери сторінок. |
| headings | Це значення за замовчуванням для класу «книга». «Нога» порожня, «голова» містить номер сторінки і назву розділу або підрозділу, в залежності від класу документа та його параметри. |
| myheadings | Інформація розділу в «голові» не вказані, але може бути вказаною явно за допомогою команди `\markright` або `\markboth`. |

Команда `\markboth` використовується з опцією `twoside`, де парними сторінками вважаються ліві, а непарними - праві. З опцією `oneside` всі сторінки вважається правими, і в даному випадку може бути використана команда `\markright`. Ці команди також можуть бути використані для перевизначення стандартних «голів», що встановлені стилями. Вони дають лише обмежений контроль над «головою» і «ногою» листа. Загальний формат, шрифти, використовування і розміщення номерів сторінок фіксуються системою LATEX [4,15].

Стиль нумерації сторінок може бути заданий за допомогою наступної команди:

`\pagenumbering{...}`

Можливі аргументи для цієї команди, і результати залучення стилю показані у таблиці 5.

Аргумент команди нумерації сторінок за умовчанням встановлюється `arabic`. Ця команда обнулює лічильник сторінок. Наприклад, якщо до числа всіх сторінок входить «Передмова» з римськими цифрами та іншою частиною документа з індійсько-арабськими цифрами, то необхідно вказати команду `\pagenumbering{roman}` на початку передмови. Після цього необхідно виконати команду `\pagestyle{arabic}` після першої команди `\chapter`, яка починає нову главу.

Допускається встановлення нумерації сторінки будь-яким числом за допомогою наступної команди:

`\setcounter{page}{`*`number`*`}`

де `number` необхідний номер для поточної сторінки.

Сторінка LATEX складається не тільки з «голови» і «ноги», але й тіла, що містить власне текст. У форматуванні сторінки система LATEX використовує ширину і висоту цих частин сторінки та інші параметри, такі як ліве і праве поля. Значення цих параметрів встановлюють розмір паперу, опції і формат сторінки та команд стилю. Наприклад, команда:

`\setlength{\textwidth}{15cm}`

встановлює ширину тексту 15 см. Додатковий пакет `geometry` надає простіший інтерфейс для

налаштування формату сторінки [23].

Таблиця 5.

**Перелік аргументів команди нумерації сторінок**

| Значення аргументів | Опис аргументів |
|---|---|
| arabic | Індо-арабська нумерація |
| roman | Римські цифри |
| Roman | Римські цифри у верхньому регістрі |
| Alph | Латинські букви |
| Alph | Латинські букви у верхньому регістрі |

**Висновок**

В роботі досліджено основні можливості та переваги використання системи Латекс для створення наукових документів. Результати дослідження показують наступні переваги LaTeX-системи як засобу форматування технічної літератури:

- висока якість створеного документа, який відповідає типографським стандартам (зручні засоби відтворення алфавітного покажчика, списків літератури, графічних об'єктів і таблиць, автоматична нумерація математичних формул, посилань та інших об'єктів поряд із ефективним механізмом перехресного цитування);
- можливість встановлення системи LaTeX на будь-якому персональному комп'ютері та безкоштовне її розповсюдження;
- можливість набору вхідного файлу у будь-якому текстовому редакторі (на відміну від файлу, створеного в редакторі Microsoft ®Word, LaTeX-файл має невеликий розмір і не потребує додаткової архівації);
- наявні засоби форматування математичних виразів;
- можливість миттєвого представлення результатів у вигляді графіків, діаграм, тощо.

**Література**

1. Knuth Donald E. The TeXbook / Donald E. Knuth. - Addison-Wesley Professional, 1984. - 496 p.
2. International Publisher Science, Technology, Medicine - Режим доступу до ресурсу : http://www.springer.com.
3. Abrahams Paul W. Tex for the Impatient / Paul W. Abrahams, Kathryn A Hargreaves, Karl Berry. - Addison Wesley Publishing Company, 2000. - 357 p.
4. Kopka H. Guide to LaTeX / Helmut Kopka , Patrick W. Daly ; [4 edition]. - Addison-Wesley Professional, 2003. - 624 p.
5. Kottwitz S. LaTeX Beginner's Guide / Stefan Kottwitz. - Packt Publishing, 2011. - 336 p.
6. Griffiths David F. Learning LaTeX / David F. Griffiths, Desmond J. Higham. - SIAM: Society for Industrial and Applied Mathematics, 1997. - 184 p.
7. Lamport L. LaTeX: A Document Preparation System / Leslie Lamport ; [2nd Edition]. - Addison-Wesley Professional, 1994. - 288 p.
8. Mittelbach F. The LaTeX Companion / Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley [Tools and Techniques for Computer Typesetting]. - Addison-Wesley Professional, 2004. – 1120 p.
9. Grätzer G. More Math Into LaTeX / George Grätzer [4 edition]. – Springer, 2007. - 654 p.
10. Довідн; За ред. Р.В. Бойка. – 3-є вид., випр.. і допов. – К.: Ред. «бюл. Вищої атестац. Коміс. України»; Вид-во «Толока», 2006. – 70 с.
11. Goossens M. The LATEX Graphics Companion / Michel Goossens, Sebastian Rahtz and Frank Mittelbach Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
12. Volovich V. Cyrillic languages support in LATEX / Vladimir Volovich, Werner Lemberg and LATEX3 Project Team Comes with the LATEX2" distribution as cyrguide.tex.
13. Режим доступу до ресурсу.: http://www.ctan.org/tex-archive/info/symbols/comprehensive/, July 2001.
14. Williams G. The TeX Catalogue is a very complete listing of many TEX and LATEX related packages / Graham Williams. - Режим доступу до ресурсу.:CTAN: //help/Catalogue/catalogue.html.
15. Reckdahl K. Using EPS Graphics in LATEX2" Documents, which explains everything and much more than you ever wanted to know about EPS files and their use in LATEX documents / Keith Reckdahl. - Режим доступу до ресурсу.: CTAN://info/epslatex.ps.
16. Kristoffer H. Rose. XY-pic User's Guide / Kristoffer H. Rose. - Режим доступу до ресурсу.: CTAN with XY-pic distribution.
17. Hobby John D. A User's Manual for METAPOST / John D. Hobby. - Режим доступу до ресурсу.: http://cm.bell-labs.com/who/hobby.
18. Hoenig A. TEX Unbound / Alan Hoenig. - Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-

509686-X (pbk).

19. Oswald U. Graphics in LATEX2", containing some Java source files for generating arbitrary circles and ellipses within the picture environment, and METAPOST / Urs Oswald [A Tutorial] . - Режим доступу до ресурсу.: http://www.ursoswald.ch.

20. Tantau T. TikZ&PGF Manual / Till Tantau. - Режим доступу до ресурсу.: CTAN://graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf.

21. Burt, John. Using poemscol for Critical Editions of Poetry / Burt, John [In:.e PracTEX Journal 3. 2005]. - Режим доступу до ресурсу.: http://www.tug.org/pracjourn/index.html.

22. Braams. LATEX2 / Braams, Johannes et al. - Режим доступу до ресурсу.: http://www.tug.org/texlive/Contents/live/texmf-dist/doc/latex/base/source2e.pdf.

23. Fenn, Jürgen. Managing Citations and Your Bibliography with BibTEX / Fenn, Jürgen [PracTEX Journal]. - Режим доступу до ресурсу.: http://www.tug.org/pracjourn/2006-4/fenn.

Надійшла 29.9.2012 р.
Рецензент: д.т.н. Поморова О.В,

УДК 681.3.04

І.А. ДИЧКА, М.В. ОНАЙ, О.В. ВАЩІЛІН
Національний технічний університет України «Київський політехнічний інститут»

# АПАРАТНА РЕАЛІЗАЦІЯ ОПЕРАТОРІВ ТА ФУНКЦІЙ В ПОЛЯХ ГАЛУА

*У статті обґрунтовано, що швидкодії при завадостійкому кодуванні даних та при виконанні криптографічних перетворень існує доцільність з метою побудови спеціалізованих апаратних засобів для виконання операцій в полях Галуа. Запропоновано два підходи для апаратної реалізації операторів та функцій в полях Галуа та докладно розглянуто побудову функціонального вузла для кожного оператора.*

*In paper it was grounded that to improve performance when noiseless coding of data and when performing cryptographic transformations there is a necessity of building specialized hardware for performing operations in Galois fields. Two approaches have been proposed for the hardware implementation of operators and functions in Galois fields and it has been considered in detail the construction of a functional node for each operator.*

Ключові слова: скінченне поле, схема цілочислового ділення, примітивний елемент.

## Вступ

Основні принципи теорії скінченних полів (які часто називають полями Галуа) як одного з розділів математики, розроблені в працях Ферма, Ейлера, Гауса, Галуа та інших видатних вчених [1– 3]. До недавнього часу теорія скінченних полів розвивалась як галузь класичної математики [3]. Але у зв'язку з розвитком завадостійкого кодування та криптографії активно розвиваються прикладні аспекти теорії [4– 7].

Обчислення у скінченних полях мають свою специфіку, і їх програмна реалізація з використанням універсальних комп'ютерів та мов програмування є не завжди ефективною з точки зору забезпечення потрібної швидкодії. Тому актуальною є проблема апаратної або апаратно-програмної реалізації обчислень у полях Галуа [8].

## Постановка задачі

Аналіз класів задач, які мають місце в завадостійкому кодуванні та при криптографічному захисті інформації, показує, що можливі два підходи до створення апаратних засобів для реалізації обчислень в полях Галуа – побудова спеціалізованих комп'ютерних систем (СКС) та побудова проблемно-орієнтованих процесорів (ПОПр).

Для реалізації зазначених обчислювальних засобів, перш за все, необхідно розглянути питання створення спеціалізованих функціональних вузлів, на основі яких слід проектувати СКС та ПОПр.

## Апаратний синтез операторів та функцій у скінченних полях

Розглянемо апаратну реалізацію операторів та функцій (табл. 1), які використовуються як функціонально завершені вузли для побудови функціональних блоків (спеціалізованих процесорів), орієнтованих на реалізацію арифметики полів Галуа.

***Оператор обчислення лишку числа*** $A$ ***за модулем*** $N_\Omega$ $(A \bmod N_\Omega)$**.** Нехай $A \in GF(N_\Omega)$, $A$ – ціле число. У загальному випадку для обчислення $A \bmod N_\Omega$ (мнемонічне позначення – $\mathrm{RES}(A)$, RESidue – лишок) слід скористатися схемою цілочислового ділення (СЦД) $A$ на $N_\Omega$ (рис. 1), результатом якого є частка і остача. Остачу позначимо $A \bmod N_\Omega$, вона є лишком числа за модулем $N_\Omega$ (часткою при цьому нехтують).

СЦД можна реалізувати мікропрограмно.